Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра Комплексной информационной безопасности

электронно-вычислительных систем (КИБЭВС)

СИСТЕМА ОБНАРУЖЕНИЯ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА НА ОСНОВЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Пояснительная записка

Студент гр. 715

\_\_\_\_\_\_\_\_ Б.Э. Данзырын

«\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Руководитель

Преподаватель кафедры КИБЭВС,

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ О.В. Пехов

«\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Преподаватель кафедры КИБЭВС,

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ Д.С. Никифоров

«\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Оглавление

[1 Введение 3](#_Toc13404044)

[2 Описание принципов работы системы 6](#_Toc13404045)

[2.1 Структура схемы 6](#_Toc13404046)

[2.2 Используемые устройства 7](#_Toc13404047)

[2.3 Программные модули устройств 12](#_Toc13404048)

[2.4 Описание модуля распознавания лица 13](#_Toc13404049)

[3 Разработка системы 15](#_Toc13404050)

[3.1 Подготовка к использованию IP-камеры и Orange Pi PC 15](#_Toc13404051)

[3.2 Подключение к рабочему столу Orange Pi PC 17](#_Toc13404052)

[4 Использование алгоритмов компьютерного зрения 24](#_Toc13404053)

[4.1 Получение кадра из видеопотока 24](#_Toc13404054)

[4.2 Распознавание лица на полученном кадре 25](#_Toc13404055)

[4.3 Установка библиотек OpenCV на Orange Pi PC 26](#_Toc13404056)

[4.4 Передача сохраненного кадра через клиент-серверное приложение 27](#_Toc13404057)

[4.5 Обучение на базе фотографий и распознавание 29](#_Toc13404058)

[4 Заключение 32](#_Toc13404059)

# 1 Введение

История современных технологий вошла в новый виток — специалисты и обозреватели называют это явление «интернет вещей».

«Интернет вещей (англ. internet of things, IoT) — концепция вычислительной сети физических предметов, оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключающее из части действий и операций необходимость участия человека» [1].

Технологии интернета вещей успешно используются практически во всех областях, например:

* транспорт: умные автомобили способны определить расстояние до ближайшей машины, оценить обстановку на дороге, проанализировать погодные условия и т.д. За счет этого повышается безопасность в грузоперевозках и безопасность для спецтехники, такси и личного транспорта;
* ЖКХ: довольно популярное направление развития IoT – «умные города». Сенсоры на объектах городской инфраструктуры – от трамваев и троллейбусов до столбов уличного освещения – позволяют следить за состоянием города. Если вы когда-нибудь хотели знать, сколько электроэнергии потребляет отдельное устройство, подключенное к сети, то это стало возможным благодаря умным счетчикам энергопотребления, которые умеют дезагрегировать энергию;
* медицина: другим направлением развития IoT стали разработки носимых медицинских устройств, позволяющих получить больше данных о состоянии здоровья пациента и проводить удаленную диагностику;
* ритейл: в сфере логистики уже давно следят за передвижениями грузовиков: учитывают использование топлива, проверяют температуру внутри рефрижератора и т.д.; следить за перемещением заказанного товара стало еще проще – от момента его упаковки до доставки лично на дому.

По сути еще до появления интернета вещей уже существовала взаимосвязь устройств класса M2M (machine2machine) коммуникаций. Но что же тогда изменилось, раз наступил скачок в развитии IoT? А изменилось то, что устройства стали общаться между собой посредством сетевых технологий Интернет, а не по специализированным системам связи.

Применительно к безопасности, такие технологии IoT избавляют нас от лишних «проводов» доводя их до минимума. Важным преимуществом таких решений является беспроводное взаимодействие устройств. В практике такие решения в безопасности становятся очень гибкими для реализации.

Например, при работе с камерами упрощается расположение и установка системы за счет беспроводного соединения устройств. IP-камеры можно установить где угодно, подключить к маршрутизатору по Wi-Fi, транслируя на определенные IP-адреса видеопоток.

Далее к этим видеопотокам можно подключить контроллеры. На контроллере видеопоток будет обрабатываться программой, обнаруживающей лица на кадре. В качестве контроллеров могут выступать одноплатные компьютеры на базе процессоров с низким энергопотреблением ARM.

Или для системы контроля и управления доступом (СКУД) в охраняемое помещение с помощью подключения IP-камер можно добавить распознавание по лицу. И таким образом избежать привлечения человеческих ресурсов, установки дополнительного оборудования в виде считывателей карт, расширить зону обнаружения за счет простой установки беспроводных IP-камер. А также добавить в функционал обнаружение несанкционированного доступа.

Простую в эксплуатации и развертывании систему обнаружения можно использовать не только в организациях, но и в частном секторе. Доступные IP-камер и одноплатные компьютеры может себе позволить почти любой среднестатистический человек и использовать у себя, например, на даче.

При этом не нужно будет тянуть множество проводов. Достаточно установить на контроллер приложение распознающее лицо из видеопотока и получать удаленный доступ к данным. Другой вариант – использовать на сервере программу, обучающуюся распознавать лица на базе фотографий и распознающую на полученном от контроллера кадре лицо человека. В качестве сервера может выступать виртуальная машина в облачном сервисе или домашний компьютер.

Целью данной работы является разработка системы обнаружения несанкционированного доступа на основе компьютерного зрения.

В процессе выполнения поставленной задачи необходимо:

1. Рассмотреть основные принципы работы системы, существующие методы распознавания образов относительно компьютерного зрения;
2. Разработать алгоритм и подробно описать предложенный подход к решению поставленной задачи;
3. Провести испытания предложенной системы обнаружения.

# 2 Описание принципов работы системы

## 2.1 Структура схемы

Для выполнения работы используются одноплатный компьютер, IP-камера, маршрутизатор и персональный компьютер, которые подключались по следующей схеме, указанной на рисунке 2.1.

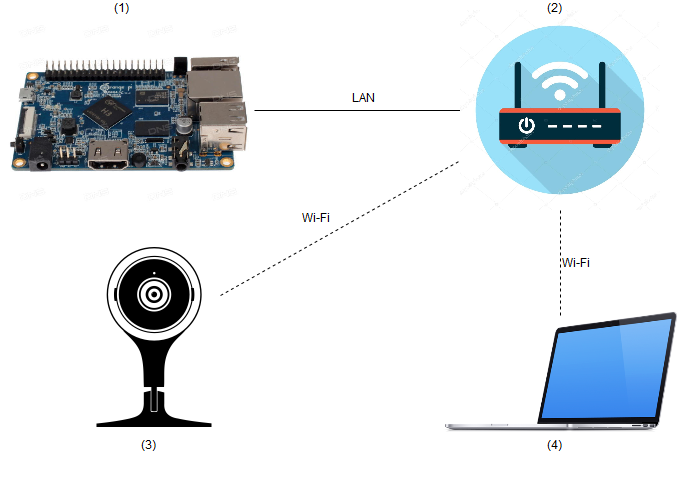


Рисунок 2.1 – Схема сетевого подключения устройств: 1 – одноплатный компьютер; 2 – маршрутизатор; 3 – IP-камера; 4 – персональный компьютер

IP-камера подключена к маршрутизатору посредством беспроводной связи Wi-Fi. При подключении ей автоматически назначается IP-адрес, по которому идет вещание видеопотока.

В роли контроллера выступает одноплатный компьютер, подключенный к маршрутизатору посредством LAN-кабеля. На нем запущено соответствующее приложение для подключения к видеопотоку, обработки кадров и обнаружения на них лиц. Затем после обработки кадры сохраняются и передаются на сервер.

Для осуществления передачи на контроллере установлено клиентское приложение, открывающее соединение с серверным приложением и закрывающее его после успешной передачи изображения.

В роли сервера выступает персональный компьютер, соединяющийся с маршрутизатором по Wi-Fi. На сервере запущена программа, которая обучается на фотографиях людей уже имеющихся в определённой папке. Затем запущенная серверное приложение ожидает запросы на подключение и передачу изображения. После получения изображение используется программой для распознавания. В результате программа выводит ответ корректно или некорректно распознано лицо и есть ли человек с таким лицом в списке имеющихся фотографий.

## 2.2 Используемые устройства

В качестве IP-камеры используется DIGMA DiVision 100 (рисунок 2.2).



Рисунок 2.2 – DIGMA DiVision 100

Сетевая IP-камера Digma DiVision 100 – небольшое устройство, которое можно устанавливать на любую ровную поверхность или подвешивать к стене. Оно не нуждается в большом количестве проводов, поскольку подключается к домашней сети с помощью встроенного адаптера Wi-Fi. Камера работает и при недостаточном свете – автоматически включается модуль ИК-посветки.

Устройство снабжено светочувствительной CMOS-матрицей, стеклянным объективом с двойным инфракрасным фильтром и инфракрасными светодиодами. Оно может получать четкую детализованную картинку в любых условиях – от яркого солнечного освещения до полной темноты.

Благодаря простому управлению и доступной цене была выбрана эта камера. Для получения доступа к видео и настройкам камеры достаточно воспользоваться удобным веб-интерфейсом или фирменным мобильным приложением. Устройство поддерживает подключение к видеопотоку по протоколу RTSP.

Характеристики IP-камеры:

1. разрешение видео – 1280x720;
2. скорость передачи видео (кадров в секунку) – 25;
3. версия встроенного ПО – A70.69.1.17.20.bin;
4. угол обзора (в градусах) – 80;
5. ИК-подсветка – да;
6. Wi-Fi – 802.11/b/g/n.

Используемые комплектующие:

1. IP-камера DIGMA DiVision 100;
2. сетевой адаптер с USB-кабелем.

В качестве одноплатного компьютера был выбран Orange Pi PC. Изображение и описание аппаратных модулей одноплатного компьютера представлено на рисунке 2.3.

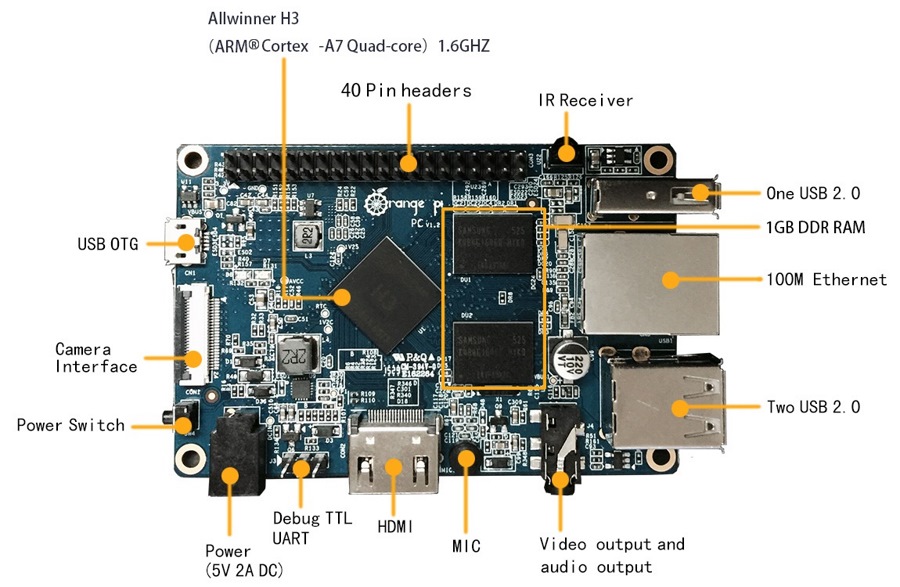


Рисунок 2.3 – аппаратные модули платы Orange Pi PC

В рамках этой платы работает программа, подключающаяся к видеопотоку с IP-камеры. В ней реализован модуль обработки кадров для обнаружения лиц. Затем после обработки кадры сохраняются и передаются на сервер. Для осуществления передачи на контроллере установлено клиентское приложение, открывающее соединение с серверным приложением и закрывающее его после успешной передачи изображения.

Устройство имеет на плате 4-х ядерный ARM процессор Allwiner H3 (Cortex-A7) с частотой 1.6Ghz и с поддержкой декодирования кодека H265/HEVC 4K, а также интегрированное видео ядро Mali400MP2 с частотой 600Mhz.Оперативная память 1Gb производства Samsung. На плате имеются порты USB 2.0 в количестве 4 штук, один порт micro-USB в качестве USB-OTG, HDMI с поддержкой HDMI-CEC, 100 мегабитный порт Ethernet и аудио-видео выход под RCA реализованный через разъем jack-3,5mm. Среди прочего на плате имеются микрофон, IR-приемник для пульта, 40 пиновый GPIO совместимый с модулями Raspberry, тактовая кнопка питания, разъем под шлейф камеры и TTL-UART пины для отладки [3].

Благодаря низкой цене при заявленных характеристиках, эта плата стала лучшей среди доступных на рынке.

Характеристики одноплатного компьютера:

1. процессор – H3 четырехъядерный процессор Cortex-A7;
2. воспроизведение видео – 4К H.265/HEVC в 30к/с, 1080р в 60к/с;
3. GPU – Mali400MP2 GPU @ 600 мГц;
4. SDRAM – 1 ГБ DDR3 (совместно с GPU);
5. объем/тип карты памяти – макс. 64 ГБ/слот карты MMC;
6. сетевой адаптер – 10/100 Ethernet RJ45;
7. поддержка HDMI – выход с поддержкой HDCP;
8. аудио выход – 3.5 мм разъем и HDMI;
9. USB 2.0 порты – 3;
10. низкоуровневая периферия – 40 контактов, совместимы с Raspberry Pi B +, GPIO (1x3), UART, ground.

Используемые комплектующие:

1. одноплатный компьютер Orange Pi PC;
2. USB-адаптер на microSD;
3. сетевой адаптер с кабелем питания.

В качестве персонального компьютера, используемом в роли сервера, используется Ноутбук 15.6” Samsung NP300V5A-S0VRU, предназначенный для бюджетного сегмента рынка.

Характеристики:

1. процессор – Intel Core i5 2430M 2 ядра 2.4 ГГц;
2. оперативная память – 6Гб;
3. модель дискретной карты – GeForce GT 520M;
4. объем HDD – 500 ГБ.

В синтетических тестах PerfomanceTEST процессор данного устройства показывает результат в 3285 ед. Вычисления в однопоточном режиме – 1363 ед. В многопоточном – 3393 ед [4].

Связующим звеном контроллера, IP-камеры и персонального компьютера выступает маршрутизатор tp-link TL-WR841ND – это комбинированное проводное/беспроводное сетевое устройство со скоростью беспроводного соединения до 300 Мбит/с, предназначенное для использования в малых и домашних офисах. Имеет удобный веб-интерфейс для управления подключениями [5].

Характеристики:

1. интерфейс – 4 порта LAN на 10/100 Мбит/с 1 порт WAN на 10/100 Мбит/с;
2. кнопки – WPS/Reset Wireless On/Off (Вкл./Выкл. беспроводное вещание) Power On/Off (Вкл./Выкл. питание);
3. внешний источник питания – 9В пост. тока / 0,6A;
4. стандарты беспроводной передачи данных – IEEE 802.11n, IEEE 802.11g, IEEE 802.11b;
5. антенна – 2\*5 дБи съемные всенаправленные антенны (RP-SMA);
6. диапазон частот (приём и передача) – 2400-2483,5 МГц;
7. скороcть передачи – 11n: до 300 Мбит/с (динамическая), 11g: до 54 Мбит/с (динамическая), 11b: до 11 Мбит/с (динамическая);
8. чувствительность (приём) – 270M: -68 дБм при 10% PER, 130M: -68 дБм при 10% PER, 108M: -68 дБм при 10% PER, 54M: -68 дБм при 10% PER, 11M: -85 дБм при 8% PER, 6M: -88 дБм при 10% PER, 1M: -90 дБм при 8% PER;
9. EIRP (Мощность беспроводного сигнала) – < 20 дБм или < 100 мВт;
10. функции беспроводного режима – Включение/выключение беспроводного вещания, WDS "мост", WMM, статистика беспроводного режима;
11. защита беспроводной сети – 64/128/152-битное шифрование WEP/WPA/WPA2, WPA-PSK/WPA2-PSK.

## 2.3 Программные модули устройств

В IP-камере DIGMA DiVision 100 используется заводская прошивка A70.69.1.17.20.bin. Для управления ее конфигурацией на смартфон устанавливается приложение DIGMA SmartCam.

Для установки операционной системы (ОС) на Orange Pi PC была выбрана Armbian 5.83 Ubuntu\_xenial\_default\_3.4.113 версии desktop. Удобством данной ОС является предустановленный графический интерфейс пользователя. Дополнительно для предоставления функциональной возможности удаленного рабочего стола установлены пакеты «tightvncserver» и «xrdp»; для разработки программы, обрабатывающей видеопоток, установлены «guvcview», «synaptic», «python-dev», «python-numpy», «python-scipy», «python-matplotlib», «python-pandas», «python-nose», «build-essential», «cmake», «pkg-config», «default-jdk», «ant», «libgtkglext1-dev», «bison», «qt4-dev-tools», «libqt4-dev», «v4l-utils», «qtcreator». Удобством данного решения является возможность компиляции программ на с++ с использованием библиотек OpenCV непосредственно на Orange Pi PC.

На маршрутизаторе tp-link TL-WR841ND используется заводская прошивка «TL-WR841ND\_V11\_Datasheet». Данная версия прошивки имеет удобный веб-интерфейс администратора.

На персональном компьютере используется ОС Windows 10 Корпоративная. Для разработки программы распознавания и клиент-серверного приложения дополнительно установлены: Visual Studio 2017, OpenCV 3.4.6, python-3.7.3, numpy, ONVIF Device Manager, rufus 2.18.1213, PuTTY, opencv-python.

В качестве клиента удаленного рабочего стола использовалось предустановленное в Windows 10 приложение «mstsc.exe» – «Подключение к удаленному рабочему столу».

## 2.4 Описание модуля распознавания лица

Исходя из поставленной задачи распознавания лица человека, необходимо выбрать метод распознавания. Из существующих методов решения распознавания образов был сделан выбор в пользу метода каскадных классификаторов Хаара. Он реализован в библиотеке OpenCV и имеет шаблоны каскадов, с которыми будет вестись работа в дальнейшем.

Как распознается лицо на кадре. Во-первых, нужно найти, где на кадре расположено лицо человека и не спутать его с картиной на стене или кошкой. Подобного рода задача становится непростой для компьютерного зрения. Для того, чтобы найти лицо необходимо выделить его основные компоненты, такие как нос, лоб, глаза и т.д. Для этого используются шаблоны, они же признаки Хаара (рисунок 2.4) [6].

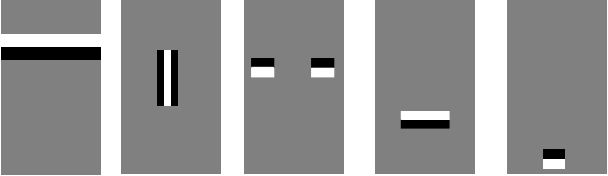


Рисунок 2.4 – Примеры шаблонов

В случае, если шаблоны соответствуют конкретным областям на изображении, считается, что на изображении есть человеческое лицо. Для каждого из них считается разность между яркостью белой и черной областей. Это значение сравнивается с эталоном и принимается решение о том, есть ли здесь часть человеческого лица или нет.

Этот метод называется методом каскадов Хаара. Что происходит, когда на фотографии не одно большое лицо, а много маленьких. Если применить шаблоны ко всей картинке, то никаких лиц не обнаружится, так как они будут меньше шаблонов. Для того чтобы искать на всем фото лица разных размеров используется метод скользящего окна. Именно внутри этого окна и высчитываются примитивы.

Окно как бы скользит по всему изображению. После каждого прохождения изображения размер окна увеличивается для поиска лиц большего масштаба.

За распознавание лица на кадре из видеопотока с помощью таблиц каскадов Хаара отвечает функция «face\_cascade.detectMultiScale()» из создаваемого объекта «face\_cascade» в классе «CascadeClassifier» библиотеки OpenCV. В данной функции при разработке используются следующие параметры: «img», «faces», «1.1», «2», «0 | CV\_HAAR\_SCALE\_IMAGE», «Size(30, 30)» [7].

Первый параметр – Матрица типа CV\_8U, содержащая изображение, в котором обнаруживаются объекты, в нее передается полученный кадр из видеопотока;

Второй параметр – Вектор прямоугольников, где каждый прямоугольник содержит обнаруженный объект;

Третий параметр – Параметр, указывающий, насколько уменьшается размер изображения при каждом масштабе изображения;

Четвертый параметр – Параметр, указывающий, сколько соседей должен иметь каждый прямоугольник – кандидат для его сохранения;

Пятый параметр – Параметр с тем же значением для старого каскада, что и в функции «cvHaarDetectObjects». Он не используется в новой версии каскада. Однако в данном параметре в целях лучшей оптимизации работы с оперативной памятью используется «CV\_HAAR\_SCALE\_IMAGE».

Шестой параметр – минимально и максимально возможный размер объекта. Объекты размера, не входящего в диапазон, игнорируются.

# 3 Разработка системы

## 3.1 Подготовка к использованию IP-камеры и Orange Pi PC

3.1.1 Подключение IP-камеры к Wi-Fi. Доступ к видеопотоку по RTSP

Для начала подключается кабель питания к IP-камере. Идет ожидание голосового сообщения «устройство готово к работе». Далее необходимо «передать» настройки Wi-Fi подключения через приложение DIGMA SmartCam разработанное для смартфонов на ОС Android.

После установки DIGMA SmartCam добавляется новое устройство. Далее идет переход в пункт «Быстрая настройка Wi-Fi». После выполнения инструкций в «Быстрой настройке», завершается подключение IP-камеры к Wi-Fi. После завершения подключения обязательно устанавливается логин и пароль для администратора камеры. По желанию можно установить логин и пароль пользователя камеры.

Прошивка камеры поддерживает стороннее подключение программ по двум протоколам HTTP и RTSP. Учитывая возможности работы с медиапотоком предпочтительнее стал RTSP.

RTSP (англ. Real Time Streaming Protocol, или, потоковый протокол реального времени) – это прикладной протокол, в котором описаны команды для управления видеопотоком. RTSP не выполняет сжатие, а также не определяет метод инкапсуляции мультимедийных данных и транспортные протоколы. Передача потоковых данных сама по себе не является частью протокола RTSP. Большинство серверов RTSP используют для этого стандартный транспортный протокол реального времени, осуществляющий передачу аудио- и видеоданных [8].

Протокол (RTSP) дает возможность клиенту запрашивать живые или предварительно записанные потоки с мультимедийных серверов, подобно тому, как HTTP позволяет клиентам выдавать запросы к Web-cepверам.

Для подключения к IP-камере по RTSP-протоколу необходимо разрешить подключение, включается соответствующая функция в разделе «Расширенные настройки», в подразделе «Настройки ONVIF\_RTSP».

Каждый производитель IP-камер самостоятельно принимает решение по какому формату подключаться к видеопотоку по RTSP. Чтобы точно узнать формат подключения используется ONVIF Device Manager.

На персональный компьютер устанавливается приложение ONVIF Device Manager. После аутентификации с использованием логина и пароля администратора IP-камеры программа автоматически определяет доступную IP-камеру из списка подключенных устройств маршрутизатора. Запустив функцию «Live Stream» можно получить соответствующий формат подключения к IP-камере по RTSP-протоколу (рисунок 3.1).

Формат подключения используется при разработке приложения.

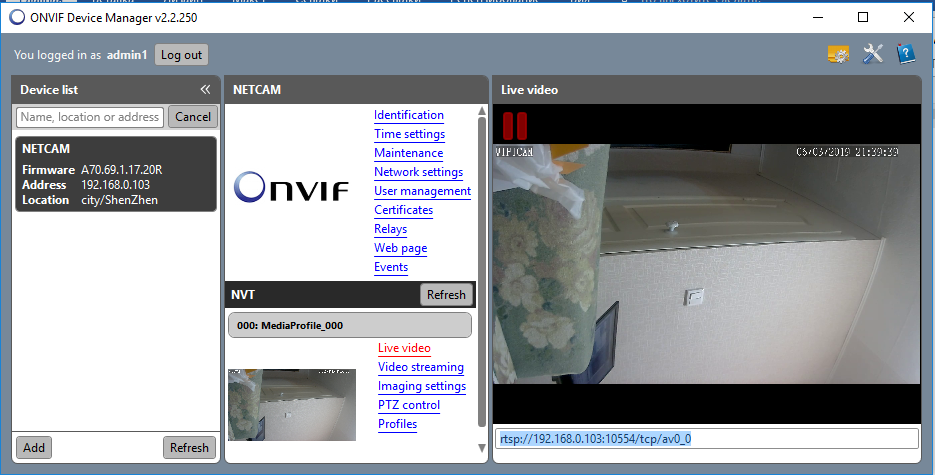


Рисунок 3.1 – Интерфейс ONVIF Device Manager

3.1.2 Установка ОС Armbian на microSD

Для дальнейшего использования Orange Pi PC требуется ОС с установленным графическим интерфейсом пользователя. Былс делан выбор в пользу Armbian 5.83 Orangepipc Ubuntu xenialdefault 3.4.113 desktop [9].

Для установки ОС на microSD используется программа rufus версии 2.18 (рисунок 3.2). После установки mircoSD помещается в выделенный разъем на Orange Pi PC.

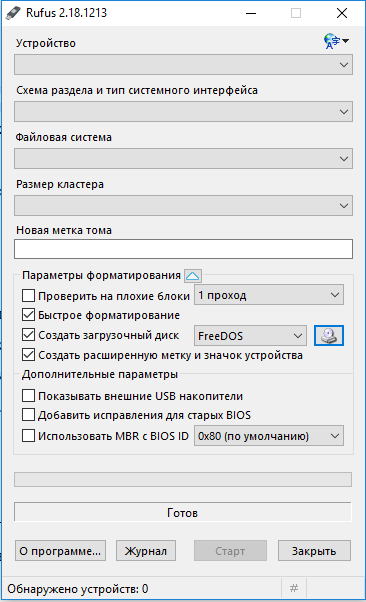


Рисунок 3.2 – Интерфейс программы rufus 2.18

## 3.2 Подключение к рабочему столу Orange Pi PC

Предустановленных возможностей удаленного рабочего стола на операционных системах для Orange Pi PC нет. Поэтому для начала необходимо подключить маршрутизатор к сети Интернет, подключиться по SSH к Orange Pi PC и начать новую сессию в терминале.

3.2.1 Подключение по SSH к Orange Pi PC

Для подключения к Orange Pi PC необходимо знать его IP-адрес. После подключения персонального компьютера и Orange Pi PC к маршрутизатору открывается интерфейс управления маршрутизатора. В меню выбирается раздел «DHCP» и в «Списке клиентов DHCP» можно узнать IP-адреса всех подключенных к маршрутизатору устройств, в том числе и Orange Pi PC (рисунок 3.3).

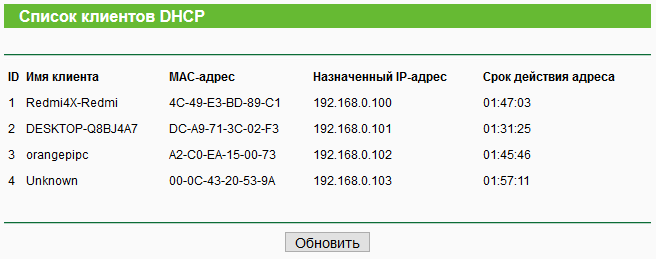


Рисунок 3.3 – IP-адрес Orange Pi PC

Для удаленного доступа используется клиент PuTTY. В поле «Host Name» вводится уже известный IP-адрес Orange Pi PC. Выбирается тип подключения SSH. По нажатию кнопки «Open» запускается новая сессия удаленного доступа по протоколу SSH (рисунки 3.4 и 3.5).

SSH (англ. Secure Shell — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений. SSH по умолчанию использует порт 22 [10].

Далее вводятся логин и пароль администратора Armbian. По умолчанию: «root» и «1234». После первой успешной аутентификации рекомендуется установить новый пароль для пользователя root.

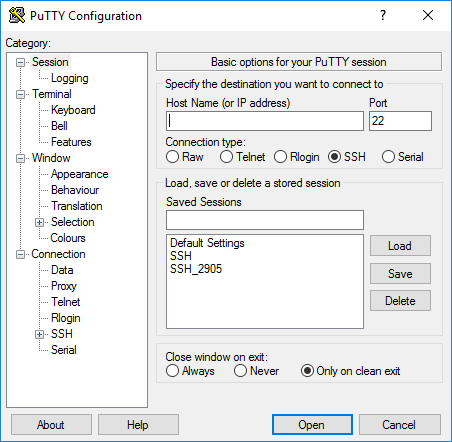


Рисунок 3.4 – Интерфейс PuTTY

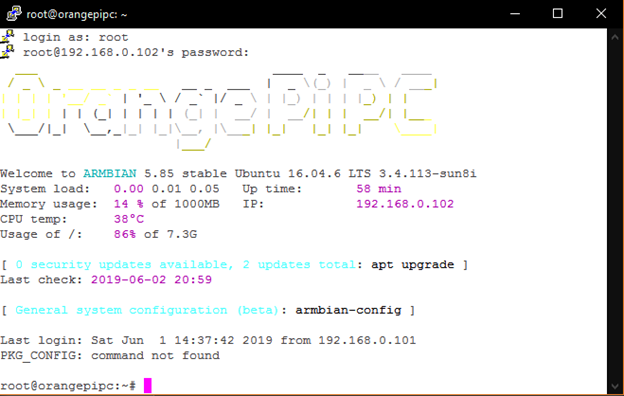


Рисунок 3.5 – Успешное подключение по SSH и запуск новой сессии в Orange Pi PC

3.2.2 Установка xrdp-сервера на Orange Pi PC для подключения к удаленному рабочему столу

Для удаленного управления рабочим столом ОС Armbian нужно установить на Orange Pi PC сервер Xrdp [11], предоставляющий функциональные возможности удаленного рабочего стола для подключаемых клиентов.

В режиме доступа по SSH в терминале выполняются следующие команды:

* «sudo apt-get purge vnc4server»;
* «sudo apt-get purge tightvnc»;
* «sudo apt-get purge xrdp»;
* «sudo apt-get install tightvncserver»;
* «sudo apt-get install xrdp».

Для выполнения данных и последующих операций по установке необходимых пакетов нужно подключение маршрутизатора к сети Интернет.

Перезагружаем устройство. Далее можно воспользоваться с уже встроенным приложением «mstsc.exe» в ОС Windows (рисунок 3.6 – 3.7) или любым другим клиентом удаленного рабочего стола. Подключение к удаленному рабочему столу идет через RDP-протокол. Результат успешного подключения показано на рисунке 3.8.

RDP (англ. Remote Desktop Protocol — протокол удалённого рабочего стола) — проприетарный протокол прикладного уровня, использующийся для обеспечения удалённого доступа пользователя с сервером, на котором запущен сервис терминальных подключений; создан компанией Microsoft. Клиенты существуют практически для всех версий Windows, Linux, FreeBSD, Mac OS X, iOS, Android, Symbian. По умолчанию используется порт TCP 3389 [12].

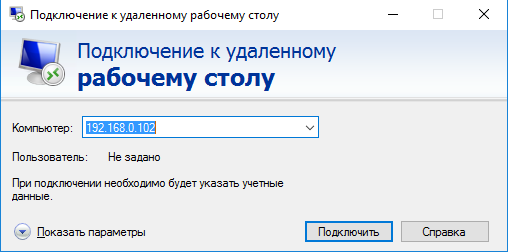


Рисунок 3.6 – Интерфейс «mstsc.exe»

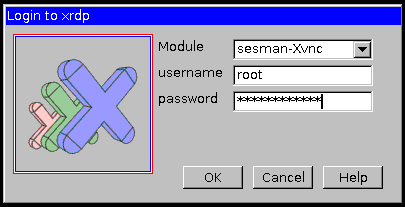


Рисунок 3.7 – Аутентификация

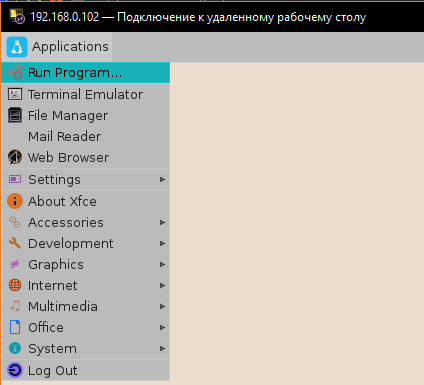


Рисунок 3.8 – Успешное подключение

3.2.3 Проверка соединения между Orange Pi PC и IP-камерой

Для проверки соединения к видеопотоку с Orange Pi PC открываем на Orange Pi PC встроенный веб браузер, переходим по адресу «http://192.168.0.102:12746» (рисунок 3.9), проходим авторизацию. Информацию о адресе подключения к веб-интерфейсу IP-камеры можно узнать из раздела «Настройки WWW» в «Расширенные настройки» устройства из списка DIGMA SmartCam.

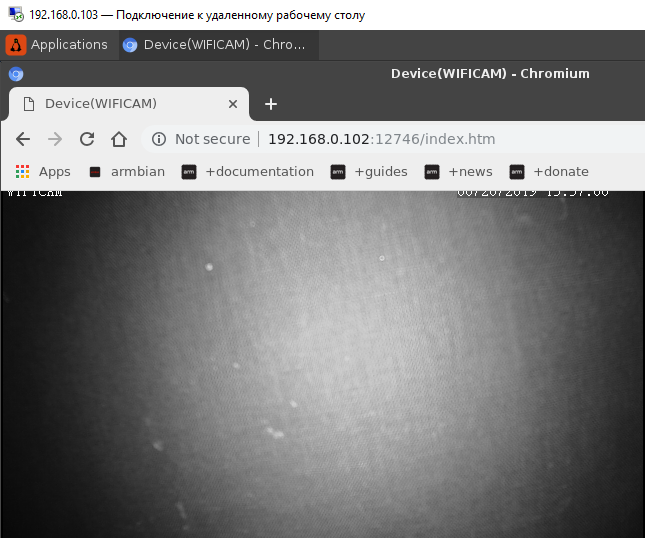


Рисунок 3.9 – Соединение есть

# 4 Использование алгоритмов компьютерного зрения

## 4.1 Получение кадра из видеопотока

Для работы с видеопотоком и распознаванием образов используются методы из библиотек компьютерного зрения OpenCV.

Создается объект «cap» типа «cv::VideoCapture». С помощью метода «cap.open(addr)» идет подключение к видеопотоку по адресу «addr». Таким образом в библиотеке OpenCV подключаются к видеопотоку IP-камеры.

Если метод возвращает не «null», создается объект «img», иначе работа программы завершается. Далее в бесконечном цикле можно обрабатывать получаемые кадры из «cap» в «img» методом «cap >> img».

Реализованный код на С++ показан на рисунке 4.1.

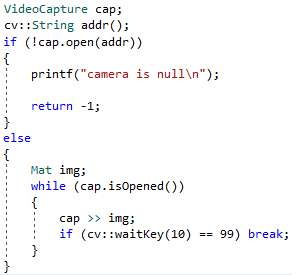


Рисунок 4.1 – Реализованный код

Примечание. Адрес подключения к IP-камере «addr» специфичный и зависит от модели и производителя используемой IP-камеры. В данной работе формат адреса уже установлен в разделе «Разработка системы».

## 4.2 Распознавание лица на полученном кадре

Для распознавания лица на кадре из видеопотока с помощью таблиц каскадов Хаара используем функцию «face\_cascade.detectMultiScale()» библиотеки OpenCV. Описание данной функции представлено в разделе «Описание принципов работы системы».

Таблицы каскадов «haarcascade\_frontalface\_alt.xml» загружаются методом «face\_cascade.load()» в объекте «face\_cascade».

Создаем объект «faces» типа «std::vector<Rect>». Этот объект используется в качестве вектора прямоугольников.

В случае заполнения вектора прямоугольников, в которых распознаны объекты, обрабатываемый кадр сохраняется методом «imwrite("result.jpg", img)», где первый параметр – это имя и формат сохранения, второй – сохраняемый объект.

Реализованный код на С++ показан на рисунке 4.2.

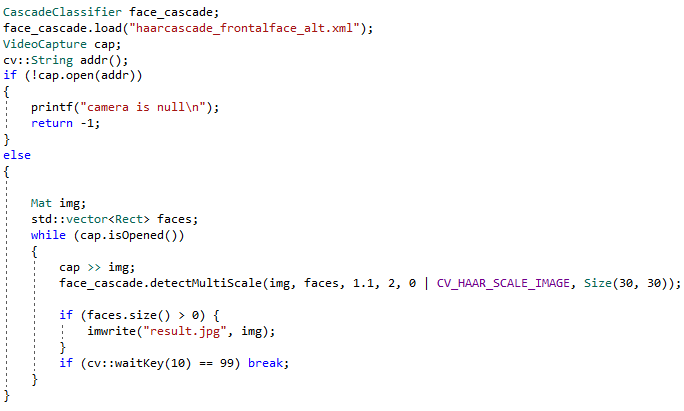


Рисунок 4.2 – Реализованный код

## 4.3 Установка библиотек OpenCV на Orange Pi PC

Применение программы для работы с видеопотоком на Orange Pi PC затрудняется необходимостью компиляции исходного кода программы на самом одноплатном компьютере. Благодаря кроссплатформенности библиотек OpenCV данное затруднение можно решить.

Для установки библиотек OpenCV на Orange Pi PC в терминале выполняются следующие команды с правами администратора, в том порядке в котором они указаны:

* «sudo apt-get update»;
* «sudo apt-get upgrade»;
* «sudo apt-get install guvcview»;
* «sudo apt-get install synaptic»;
* «sudo apt-get install python-dev»;
* «sudo apt-get install python-numpy python-scipy python-matplotlib python-pandas python-nose»;
* «sudo apt-get install build-essential cmake pkg-config»;
* «sudo apt-get install default-jdk ant»;
* «sudo apt-get install libgtkglext1-dev»;
* «sudo apt-get install bison»;
* «sudo apt-get install qt4-dev-tools libqt4-dev»;
* «sudo apt-get install v4l-utils»;
* «sudo apt-get install qtcreator»;
* «sudo wget https://sourceforge.net/projects/opencvlibrary/files/opencv-unix/3.3.5/opencv-3.4.5.zip»;
* «sudo unzip opencv-3.1.0.zip»;
* «cd opencv-3.1.0»;
* «sudo mkdir build»;
* «cd build»;
* «sudo cmake -D CMAKE\_BUILD\_TYPE = RELEASE -D INSTALL\_C\_EXAMPLES = ON –D INSTALL\_PYTHON\_EXAMPLES = ON -D BUILD\_EXAMPLES = ON -D WITH\_QT = ON -D CMAKE\_INSTALL\_PREFIX = /usr/local -D WITH\_OPENGL = ON -D WITH\_V4L = ON –D BUILD\_NEW\_PYTHON\_SUPPORT = ON -D WITH\_TBB = ON ..»;
* «sudo make»;
* «sudo make install»;
* «sudo nano /etc/ld.so.conf.d/opencv.conf».

В данном файле opencv.conf добавляется строка (рисунок 4.3).



Рисунок 4.3 – Добавляемая строка

Изменения файла сохраняются и выполняются дальнейшие команды:

* «sudo ldconfig»;
* «sudo nano /etc/bash.bashrc».

В данном файле bash.bashrc в самый конец файла добавляется строка (рисунок 4.4).



Рисунок 4.4 – Добавляемая строка

Изменения файла сохраняются и выполняется команда: «sudo export PKG\_CONFIG\_PATH».

На этом установка библиотек OpenCV на Orange Pi PC завершается и исходный код программы можно компилировать с помощью компилятора g++.

## 4.4 Передача сохраненного кадра через клиент-серверное приложение

Сохраненный кадр после обработки видеопотока необходимо передать из Orange Pi PC на персональный компьютер. Для этого на языке python разработаны серверное и клиентское приложения, использующие соответственно серверный и клиентский сокеты. Серверное приложение ожидает запросы на подключение, а клиентское, в случае сохранения изображения, отправляет запрос на подключение и передает это изображение. Благодаря кроссплатформенности данного языка программирования, нет необходимости в установке дополнительных модулей, кроме основных – python и pip.

Для клиентского приложения задается IP-адрес и его порт для подключения (рисунок 4.5)



Рисунок 4.5 – IP-адрес и порт

Создается объект «sock» типа «socket» [13] и делается попытка подключения к указанному IP-адресу и порту (рисунок 4.6)



Рисунок 4.6 – Подключение

Создается объект image и записывается в него передаваемое изображение методом «imread()» из модуля cv2 (рисунок 4.7).



Рисунок 4.7 – Запись изображения в объект

Изображение кодируется в строковый тип данных (рисунок 4.8).

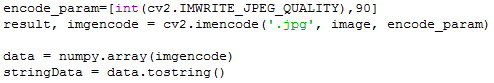


Рисунок 4.8 – Кодирование

Строка данных отправляется на сервер и затем соединение закрывается (рисунок 4.9).



Рисунок 4.9 – Отправка данных

Для серверного приложения задается IP-адрес и его порт для прослушивания (рисунок 4.10)



Рисунок 4.10 – IP-адрес и порт

Создается объект «s» типа «socket» и начинается прослушивание указанного IP-адреса и порта (рисунок 4.11).



Рисунок 4.11 – Прослушивание

В случае, когда приходит запрос на подключение, программа подключается и получает всю передаваемую строку данных. После получения данные декодируются их из строкового типа обратно в изображение и закрывается соединение (рисунок 4.12 – 4.13).

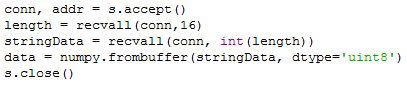


Рисунок 4.12 – Получение данных



Рисунок 4.13 – Декодирование

## 4.5 Обучение на базе фотографий и распознавание

Метод распознавания используется тот же, что и в предыдущем разделе. Однако дальше нам необходимо «узнавать» лицо. То есть использовать методы обучения программы и на основе этого принимать решение, знакомо лицо или нет.

Для детектирования лиц используются готовые таблицы значений каскад Хаара (рисунок 4.14).



Рисунок 4.14 – Подключение таблиц

Для обучения и распознавания создается объект «recognizer» типа «face» из модуля «LBPHFaceRecognizer\_create» [15], использующий локальные бинарные шаблоны (рисунок 4.15.).



Рисунок 4.15 – Локальные бинарные шаблоны

Первые два значения 1 и 8 характеризуют окрестности пикселя. Следующие параметры (8,8) характеризуют размеры областей, на которые мы разбиваем исходное изображение с лицом. Параметр 123 (confidence threshold) определяет пороговое значение для распознавания лица. Чем меньше «confidence» тем больше алгоритм уверен в том, что на фотографии изображено известное ему лицо. В данном случае порог – 123.

Обучение программы на распознавание лиц запускается методом train объекта recognizer (рисунок 4.16).



Рисунок 4.16 – Запуск обучения

Первый параметр – объекты, на основе которых обучается программа; второй параметр – их нумерация.

Далее программа получает на вход изображение, в котором необходимо найти лицо и распознать – знакомо ему или нет. В случае, если распознается лицо, то запускается цикл.

Реализованный код на python показан на рисунке 4.17.

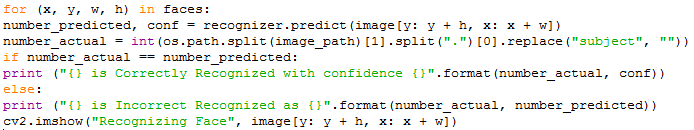


Рисунок 4.17 – Реализованный код

Функция «predict()» объекта «recognizer» возвращает номер и идентификатор субъекта, который предположительно находится на фото, а так же параметр «confidence». Далее сравнивается значение, которое нам вернула функция с номером субъекта, если они равны, распознавание прошло успешно

# 4 Заключение

В рамках данной работы была разработана система обнаружения несанкционированного доступа на основе компьютерного зрения. Были использованы: одноплатный компьютер Orange Pi PC, IP-камера Digma DiVision 100, маршрутизатор tp-link TL-WR841ND, ноутбук 15.6” Samsung NP300V5A-S0VRU.

На Orange Pi PC установлен дистрибутив Armbian 5.83 Orangepipc Ubuntu xenialdefault 3.4.113 desktop.

Разработана программа для обработки видеопотока и обнаружения лица на кадре с использованием библиотек компьютерного зрения OpenCV. Разработано клиент-серверное приложение для передачи изображения по сети. Разработана программа для распознавания лица из кадра на базе имеющихся фотографий лиц с использованием библиотек компьютерного зрения OpenCV.

Пояснительная записка была написана согласно ОС ТУСУР 01-2013.

Исходные коды представлены на: https://github.com/Rufwame/IoT